

---

# Matematikk 1000

## Øvingsoppgaver i numerikk – leksjon 7

### *Newton's metode*

---

I dette settet skal vi implementere Newtons metode. Denne algoritmen gjør essensielt det samme som halveringsmetoden – men langt mer effektivt. Den er også enklere å implementere. Så kan man kanskje lure på hvorfor vi brukte tid på halveringsmetoden i det hele. Det er flere grunner til det. En av dem bør bli ganske tydelig i oppgava nedenfor...

Vi har også lagt til ei ekstraoppgave til slutt for spesielt interesserte.

### Oppgave 1 – Newtons metode

Newtons metode går ut på å løse likninga  $f(x) = 0$  ved å iterere på rekursjonsformelen

$$x_{n+1} = x_n - f(x_n)/f'(x_n) .$$

Denne rekursjonsformelen, eller *differenslikninga*, har vi kommet fram til ved å ta utgangspunkt i nullpunktet for Taylor-polynomet av 1. orden omkring  $x_n$ .

Når du skal skrive dette skriptet, kan det kanskje være lurt å se litt på implementeringa av halveringsmetoden, som vi hadde om i leksjon 5. Det kan også være en god ide å innføre to variable: Den gamle og den nye  $x - x_n$  og  $x_{n+1}$ . Du kan for eksempel kalle dem “**xGammel**” og bare “**x**”. Algoritmen kan settes opp slik:

- 1) Velg deg en  $x_0$  som du forventer skal ligge i nærheten av løsninga.
- 2) Bestem hvor nøyaktig løsning du vil ha – hvor stor feil du tillater – og tilordne denne verdien til en variabel. Du kan for eksempel kalle denne tillatte feilen **Pres**.
- 3) Tilordne en eller annen verdi til den gamle  $x$ , **xGammel**, og la din “nye”  $x$ -verdi, **x**, være  $x_0$ . Pass på at du velger ganske ulike verdier for den gamle

og den nye  $x$ .

- 4) Gjenta følgende så lenge absoluttverdien av differansen mellom den nye og gamle  $x$ -verdien er større enn nøyaktigheten vi valgte i punkt 2:
  - 4a) La din gamle  $x$  bli det som var den nye  $x$  (`xGammel=x`).
  - 4b) La din nye  $x$  bli gitt ved formelen over (`'x=x-...'`).
- 5) Skriv den endelige  $x$ -verdien til skjerm.

Vi tar utgangspunkt i den samme likninga som i leksjon 5:

$$f(x) = 0 \quad \text{der} \quad f(x) = x - \cos x \quad .$$

- a) Løs denne likninga ved å lage et skript som implementerer Newtons metode. Velg selv en passende  $x_0$  og nøyaktighet (`Pres`). Om det går tregt å komme i gang, ikke nøl med å be om hjelp – enten fra lærer eller fra medstudenter. Ser du hvorfor det er viktig at `x` og `xGammel` i utgangspunktet blir gitt verdier som er ganske ulike?
- b) Forsøk å utvide koden slik at den også teller antall *iterasjoner* – altså hvor mange ganger punkt 4 gjentas. Undersøk hvordan dette antallet varierer med nøyaktigheten du velger. Hvordan er antall iterasjoner her sammenlignet med halveringsmetoden?
- c) Undersøk også hva som skjer når du velger en  $x_0$  i nærheten av  $3\pi/2$ .
- d) Hvor mange nullpunkter ser funksjonen  $g(x) = x^3 + 2x^2 - x - 2 + e^x$  ut til å ha? Prøv å finne disse ved å bruke Newtons metode med forskjellige verdier for  $x_0$ . Hva skjer når vi velger  $x_0 = 0$  eller  $x_0$  nær 0? Ut fra blant annet dette: Ser du noen ufordeler med Newtons metode som halveringsmetoden ikke har?

## Oppgave 2 – Mer Newtons metode

Bruk skriptet vi lagde i oppgave 1 – med visse justeringer – til å gjøre deloppgaver fra oppgave 2 (utenom d), som vi alt har gjort) og 6 fra delkappittel 3.4 i læreboka. (Gjør gjerne alle.)

## Ekstra: Oppgave 3 – Attraktive fikspunkter

For ordens skyld: Denne metoden er *ikke* pensum.

Så langt har vi sett to iterative skjema, eller algoritmer, for å løse likninger: halveringsmetoden og Newtons metode. En annen metode er *sekantmetoden*, som ligner veldig på Newtons metode – bortsett fra at den ikke involverer derivasjon direkte. Her skal vi se på en fjerde metode:

Vi tenker oss at en likning er skrevet på forma

$$x = f(x)$$

for en eller annen funksjon  $f$ . Oppgave i oppgave 1 a) er allerede på denne forma. Andre ganger må vi kanskje manipulere likninga litt for at den skal ha denne forma<sup>1</sup>. En  $x$ -verdi som oppfyller dette, kaller vi et *fikspunkt* for funksjonen  $f$ .

Om vi tar utgangspunkt i denne rekursjonsformelen:

$$x_{n+1} = f(x_n) \quad ,$$

kan vi jo håpe at  $x_n$  *konvergerer* mot en bestemt verdi – altså at  $x_{n+1} \approx x_n$  når  $n$  blir stor. Om så er tilfelle, har vi jo funnet ei løsnings. Som med Newtons metode, må vi først velge en  $x_0$  her. Om denne metoden faktisk konvergerer, er avhengig av  $x_0$  og  $f$ .

Forsøk å implementere denne metoden, og bruk den til å løse noen av de likningene vi gjorde i oppgave 1 og 2. (Du trenger bare gjøre ei helt minimal justering på skriptet for Newtons metode.) Om ting ikke konvergerer, kan du bli nødt til å bruke **Ctrl + C**-tasten for å stoppe skriptet.

---

<sup>1</sup>Oftentimes kan dette gjøres på flere måter. Og det er slett ikke sikkert at de ulike måtene å formulere likninga på gir like gode resultater når denne metoden skal implementeres.